UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/523,690	02/03/2005	Kazunori Saito	1560-0422PUS1	8523

2292 7590 10/08/2009

BIRCH STEWART KOLASCH & BIRCH
PO BOX 747
FALLS CHURCH, VA 22040-0747

EXAMINER	
SCHWARTZ, DARREN B	

ART UNIT	PAPER NUMBER
2435	

NOTIFICATION DATE	DELIVERY MODE
10/08/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

mailroom@bskb.com

Office Action Summary	Application No.	Applicant(s)	
	10/523,690	SAITO, KAZUNORI	
	Examiner	Art Unit	
	DARREN SCHWARTZ	2435	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 July 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4,5 and 7-11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4,5 and 7-11* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Applicant amends claims 1, 2, 4, 5 and 7-10.

Claims 1-5 and 7-11 are presented for examination.

Response to Arguments

Applicant's arguments with respect to claims 1-5 and 7-11 have been considered but are moot in view of the new ground(s) of rejection.

Applicant amends claims to include "reading one byte of the data; judging whether a branch destination address associated with a branch destination is larger than a branch origin address based only on the one byte of the data read; storing the branch origin address associated with the retrieved instruction code and the branch destination address associated with the branch destination of the instruction code when the branch destination address associated with the branch destination is judged to be larger than the branch origin address." The Examiner introduces Yoshimi (U.S. Pat Pub 2001/0011346 A1) *infra*.

The fact that the Examiner may not have specifically responded to any particular arguments made by Applicant and Applicant's Representative, should not be construed as indicating Examiner's agreement therewith.

Claim Objections

Claim 3 is objected to because of the following informalities:

Claim 3 is objected as the claim is identified as "Currently amended" yet no amendments are present in the most recent claim dissemination.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. Claims 1-5 and 7-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sakai et al (JP 09-128264 A), hereinafter referred to as Sakai, in view of Hollander et al (U.S. Pat 6301699 B1), hereinafter referred to as Hollander, in further view of Yoshimi, (U.S. Pat Pub 2001/0011346 A1), hereinafter referred to as Yoshimi. A translated copy of Sakai was provided in earlier correspondence.

Re claims 1, 2, 4-7: Sakai teaches a data processing method including receiving input data containing a plurality of instruction codes, said method comprising:

retrieving an instruction code related to a branch instruction from the data (page 18, lines 12-17);

judging whether or not an instruction code for calling an instruction code group for executing a predetermined process is associated with the branch destination address (page 26: ¶18; page 27, lines 8-15; page 29, lines 3-12);

storing a call destination address of the instruction code if the instruction code is associated with the branch destination address (page 3, lines 1-2; page 9, line 21 -

page 10, line 5; page 17, lines 21-23; page 22, lines 8-16; page 25, lines 4-13; page 44, see register); and

judging whether or not the stored call destination address is between the branch origin address and the branch destination address (page 9, line 21 - page 10, line 5; page 17, lines 21-23; page 22, lines 8-16; page 25, lines 4-13);

Hollander teaches:

judging whether or not a process executed based on the instruction codes contained in the received data is a malicious process (Fig 4B, all elts: col 4, lines 62-65).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai with the teachings of Hollander, for the purpose of detecting hostile executable code. Both references are within the realm of the claimed invention as both references are directed to tracing the execution of computer code.

The Examiner holds that the branch origin address and branch destination address within a computer program need not have a specific order within a computer program. It is known in the art of computer programming, that function calls could precede the currently executed statement; such practice is common in code which has been obfuscated/scrambled and/or the initial point of execution is obscured as is commonplace in polymorphic and metamorphic code. Ergo, the examiner has interpreted the limitation "judging whether or not the stored call destination address is

between the branch origin address and the branch destination address" to mean analyzing any code in an executable program.

The combination of Sakai and Hollander teaches concluding that the process executed based on the instruction codes contained in the data is a malicious process (Hollander: col 1, line 64 – col 2, line 3; Fig 4B, all elts: col 4, lines 62-65), when the instruction code for calling the instruction code group for executing the predetermined process is associated with the branch destination address and the call destination address of the instruction code is between the branch origin address and the branch destination address (Sakai: page 9, line 21 - page 10, line 5; page 17, lines 21-23; page 22, lines 8-16; page 25, lines 4-13).

While the combination of Sakai and Hollander teaches reading a plurality of instructions (Sakai: claim 1, line 6; page 17, ¶12, line 7; Hollander: Fig 7, elt 141). The combination of Sakai and Hollander is silent as to teaching judging whether a branch destination address associated with a branch destination is larger than a branch origin address based only on the one byte of the data read; storing the branch origin address associated with the retrieved instruction code and the branch destination address associated with the branch destination of the instruction code when the branch destination address associated with the branch destination is judged to be larger than the branch origin address.

Yet, Yoshimi teaches reading one byte of the data (Fig 3; ¶20; ¶23; ¶30);

judging whether a branch destination address [¶167: e.g. "branch destination address"] associated with a branch destination [¶167: e.g. "branch instruction

information"] is larger than a branch origin address [¶172: e.g. "program count value (address)"] based only on the one byte of the data read (¶172-¶173);

storing the branch origin address [¶41: "A branch instruction, a program count value and branch history information are provided …"] associated with the retrieved instruction code [Fig 3; ¶21-¶24; e.g. "instruction"] and the branch destination address [¶167: e.g. "branch destination address"] associated with the branch destination [¶167: e.g. "branch instruction information"] of the instruction code when the branch destination address [¶167: e.g. "branch destination address"] associated with the branch destination [¶167: e.g. "branch instruction information"] is judged [¶170: "comparing circuit"] to be larger than the branch origin address [¶172: e.g. "program count value (address)"] (Fig 14; ¶158; ¶170-¶176; ¶184).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai and Hollander with the teachings of Yoshimi, for the purpose of predicting program execution in file processes as taught by (¶282).

Re claim 3: The combination of Sakai, Hollander and Yoshimi teaches means for judging whether or not a predetermined character string is associated with a return address of the instruction code group, wherein if the character string is associated with the return address, the information indicating that the data is data for executing a malicious process is outputted (Sakai: pages 37 and 40: "CALL and RET instruction detecting parts;" page 42: "Branch origin/destination registers;" Hollander: Fig 4B, all elts).

Re claim 8: Sakai teaches a data processor comprising:

an input unit for inputting data containing a plurality of instruction codes (page 2, lines 1-2);

a storing unit for storing the data input by the input unit (page 2, lines 1-2); and a controller capable of performing operations (page 2, lines 1-2) of:

retrieving an instruction code related to a branch instruction from the data stored in the storing unit (page 18, lines 12-17);

judging whether or not an instruction code for calling an instruction code group for executing a predetermined process is associated with the branch destination address (page 26: ¶18; page 27, lines 8-15; page 29, lines 3-12);

storing a call destination address of the instruction code in the storing unit if the instruction code is associated with the branch destination address (page 3, lines 1-2; page 9, line 21 - page 10, line 5; page 17, lines 21-23; page 22, lines 8-16; page 25, lines 4-13; page 44, see register);

judging whether or not the stored call destination address is between the branch origin address and the branch destination address (page 9, line 21 - page 10, line 5; page 17, lines 21-23; page 22, lines 8-16; page 25, lines 4-13).

Hollander teaches:

concluding that the process executed based on the instruction codes contained in the data is a malicious process (Fig 4B, all elts: col 4, lines 62-65).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai with the teachings of

Hollander, for the purpose of detecting hostile executable code. Both references are within the realm of the claimed invention as both references are directed to tracing the execution of computer code.

The Examiner holds that the branch origin address and branch destination address within a computer program need not have a specific order within a computer program. It is known in the art of computer programming, that function calls could precede the currently executed statement; such practice is common in code which has been obfuscated/scrambled and/or the initial point of execution is obscured as is commonplace in polymorphic and metamorphic code. Ergo, the examiner has interpreted the limitation "judging whether or not the stored call destination address is between the branch origin address and the branch destination address" to mean analyzing any code in an executable program.

The combination of Sakai and Hollander teaches the instruction code for calling the instruction code group for executing the predetermined process is associated with the branch destination address and the call destination address of the instruction code is between the branch origin address and the branch destination address (Sakai: page 9, line 21 - page 10, line 5; page 17, lines 21-23; page 22, lines 8-16; page 25, lines 4-13).

While the combination of Sakai and Hollander teaches reading a plurality of instructions (Sakai: claim 1, line 6; page 17, ¶12, line 7; Hollander: Fig 7, elt 141). The combination of Sakai and Hollander is silent as to teaching judging whether a branch destination address associated with a branch destination is larger than a branch origin

address based only on the one byte of the data read; storing the branch origin address associated with the retrieved instruction code and the branch destination address associated with the branch destination of the instruction code in the storing unit when the branch destination address associated with the branch destination is judged to be larger than the branch origin address.

Yet, Yoshimi teaches reading one byte of the data (Fig 3; ¶20; ¶23; ¶30);

judging whether a branch destination address [¶167: e.g. "branch destination address"] associated with a branch destination [¶167: e.g. "branch instruction information"] is larger than a branch origin address [¶172: e.g. "program count value (address)"] based only on the one byte of the data read (¶172-¶173);

storing the branch origin address [¶41: "A branch instruction, a program count value and branch history information are provided …"] associated with the retrieved instruction code [Fig 3; ¶21-¶24; e.g. "instruction"] and the branch destination address [¶167: e.g. "branch destination address"] associated with the branch destination [¶167: e.g. "branch instruction information"] of the instruction code in the storing unit [¶23: e.g. "secondary cache;" ¶40: "branch prediction information storage part"] when the branch destination address [¶167: e.g. "branch destination address"] associated with the branch destination [¶167: e.g. "branch instruction information"] is judged [¶170: "comparing circuit"] to be larger than the branch origin address [¶172: e.g. "program count value (address)"] (Fig 14; ¶158; ¶170-¶176; ¶184).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai and Hollander with the

teachings of Yoshimi, for the purpose of predicting program execution in file processes as taught by (¶282).

Re claim 9: Sakai teaches a data processor comprising:

an input unit for inputting data containing a plurality of instruction codes (page 2, lines 1-2);

a storing unit for storing the data input by the input unit (page 2, lines 1-2); and

a controller capable of performing operations (page 2, lines 1-2) of:

retrieving an instruction code for calling an instruction code group for executing a predetermined process from the data (page 18, lines 12-17).

Hollander teaches:

concluding that the process executed based on the instruction codes contained in the data is a malicious process (Fig 4B, all elts: col 4, lines 62-65).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai with the teachings of Hollander, for the purpose of detecting hostile executable code. Both references are within the realm of the claimed invention as both references are directed to tracing the execution of computer code.

The Examiner holds that the branch origin address and branch destination address within a computer program need not have a specific order within a computer program. It is known in the art of computer programming, that function calls could precede the currently executed statement; such practice is common in code which has been obfuscated/scrambled and/or the initial point of execution is obscured as is

commonplace in polymorphic and metamorphic code. Ergo, the examiner has interpreted the limitation "judging whether or not the stored call destination address is between the branch origin address and the branch destination address" to mean analyzing any code in an executable program.

The combination of Sakai and Hollander teaches judging whether or not a predetermined character string is associated with a return address of the instruction code group (Sakai: pages 42-43, elts: ST32 & ST35).

concluding that the process executed based on the instruction codes contained in the data is a malicious process when the instruction code for calling the instruction code group for executing the predetermined process is in the data and the predetermined character string is associated with the return address of the instruction code group (Hollander: Fig 3, all elts: col 4, lines 57-58; col 5, lines 12-16).

While the combination of Sakai and Hollander teaches reading a plurality of instructions (Sakai: claim 1, line 6; page 17, ¶12, line 7; Hollander: Fig 7, elt 141). The combination of Sakai and Hollander is silent as to teaching judging whether a branch destination address associated with a branch destination is larger than a branch origin address based only on the one byte of the data read; storing the branch origin address associated with the retrieved instruction code and the branch destination address associated with the branch destination of the instruction code when the branch destination address associated with the branch destination is judged to be larger than the branch origin address.

Yet, Yoshimi teaches reading one byte of the data (Fig 3; ¶20; ¶23; ¶30);

judging whether a branch destination address [¶167: e.g. “branch destination address”] associated with a branch destination [¶167: e.g. “branch instruction information”] is larger than a branch origin address [¶172: e.g. “program count value (address)”] based only on the one byte of the data read (¶172-¶173);

storing the branch origin address [¶41: “A branch instruction, a program count value and branch history information are provided …”] associated with the retrieved instruction code [Fig 3; ¶21-¶24; e.g. “instruction”] and the branch destination address [¶167: e.g. “branch destination address”] associated with the branch destination [¶167: e.g. “branch instruction information”] of the instruction code when the branch destination address [¶167: e.g. “branch destination address”] associated with the branch destination [¶167: e.g. “branch instruction information”] is judged [¶170: “comparing circuit”] to be larger than the branch origin address [¶172: e.g. “program count value (address)”] (Fig 14; ¶158; ¶170-¶176; ¶184).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai and Hollander with the teachings of Yoshimi, for the purpose of predicting program execution in file processes as taught by (¶282).

Re claim 10: Sakai teaches a data processor comprising:

an input unit for inputting data containing a plurality of instruction codes (page 2, lines 1-2);

a storing unit for storing the data input by the input unit (page 2, lines 1-2); and

a controller capable of performing operations (page 2, lines 1-2) of:

retrieving an instruction code for calling an instruction code group for executing a predetermined process from the data (page 26: ¶18; page 27, lines 8-15; page 29, lines 3-12);

judging whether or not an instruction code for obtaining a return address of the instruction code group is contained in the instruction code group if the instruction code is retrieved (page 6: eighth step; page 19: ¶13; page 41: ST2; page 44: 7).

Hollander teaches:

concluding that the process executed based on the instruction codes contained in the data is a malicious process (Fig 4B, all elts: col 4, lines 62-65).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai with the teachings of Hollander, for the purpose of detecting hostile executable code. Both references are within the realm of the claimed invention as both references are directed to tracing the execution of computer code.

The Examiner holds that the branch origin address and branch destination address within a computer program need not have a specific order within a computer program. It is known in the art of computer programming, that function calls could precede the currently executed statement; such practice is common in code which has been obfuscated/scrambled and/or the initial point of execution is obscured as is commonplace in polymorphic and metamorphic code. Ergo, the examiner has interpreted the limitation "judging whether or not the stored call destination address is

between the branch origin address and the branch destination address" to mean analyzing any code in an executable program.

The combination of Sakai and Hollander teaches when the instruction code for calling the instruction code group for executing the predetermined process is in the data and the instruction code for obtaining the return address of the instruction code group is contained in the instruction code group (Sakai: page 9, line 21 - page 10, line 5; page 17, lines 21-23; page 22, lines 8-16; page 25, lines 4-13).

While the combination of Sakai and Hollander teaches reading a plurality of instructions (Sakai: claim 1, line 6; page 17, ¶12, line 7; Hollander: Fig 7, elt 141). The combination of Sakai and Hollander is silent as to teaching judging whether a branch destination address associated with a branch destination is larger than a branch origin address based only on the one byte of the data read; storing the branch origin address associated with the retrieved instruction code and the branch destination address associated with the branch destination of the instruction code when the branch destination address associated with the branch destination is judged to be larger than the branch origin address.

Yet, Yoshimi teaches reading one byte of the data (Fig 3; ¶20; ¶23; ¶30);

judging whether a branch destination address [¶167: e.g. "branch destination address"] associated with a branch destination [¶167: e.g. "branch instruction information"] is larger than a branch origin address [¶172: e.g. "program count value (address)"] based only on the one byte of the data read (¶172-¶173);

storing the branch origin address [¶41: "A branch instruction, a program count value and branch history information are provided …"] associated with the retrieved instruction code [Fig 3; ¶21-¶24; e.g. "instruction"] and the branch destination address [¶167: e.g. "branch destination address"] associated with the branch destination [¶167: e.g. "branch instruction information"] of the instruction code when the branch destination address [¶167: e.g. "branch destination address"] associated with the branch destination [¶167: e.g. "branch instruction information"] is judged [¶170: "comparing circuit"] to be larger than the branch origin address [¶172: e.g. "program count value (address)"] (Fig 14; ¶158; ¶170-¶176; ¶184).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Sakai and Hollander with the teachings of Yoshimi, for the purpose of predicting program execution in file processes as taught by (¶282).

Re claim 11: The combination of Sakai, Hollander and Yoshimi teaches the malicious process causes an erroneous operation in the process executed based on the instruction codes contained in the received data (Hollander: Fig 4b, elt 106: col 4, lines 62-65).

Conclusion

Examiner's Note: Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to

specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the text of the passage taught by the prior art or disclosed by the examiner.

In the case of amending the claimed invention, Applicant is respectfully requested to indicate the portion(s) of the specification which dictate(s) the structure relied on for proper interpretation and also to verify and ascertain the metes and bounds of the claimed invention.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DARREN SCHWARTZ whose telephone number is (571)270-3850. The examiner can normally be reached on 7am-4pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu can be reached on (571)272-3859. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/D. S./
Examiner, Art Unit 2435
/Kimyen Vu/
Supervisory Patent Examiner, Art Unit 2435